Amendments to the Claims

1-10 (canceled).

11 (currently amended). In a multi-tasking computer system comprising a plurality of resources to be shared by a plurality of tasks, a circuit for allocating each resource to the tasks in a continuous operation so that in said operation, after any one of the tasks has finished accessing any one of the resources in processing a data unit, said one of the tasks does not get access to the same resource until after every other one of the tasks has finished accessing the resource;

wherein accessing any one of said resources by any one of said tasks comprises:

(1) the task attempting to access the resource, wherein attempting to access the resource comprises generating a signal indicating that the task is attempting to access the resource;

(2) in response to the operation (1), the circuit allowing the task to access the resource if the resource is available to the task, the circuit not allowing the task to access the resource until the resource becomes available to the task;

(3) the task accessing the resource when the circuit allows the task to access the resource;

wherein for any task T1 of said tasks and any resource R1 of said resources, if the task T1 attempts to access the resource R1 after the task T1 has already finished accessing the resource R1, and at least one other task T2 has not attempted to access the resource R1 after the task T1 has finished accessing the resource R1, then the circuit ~~does not~~ will never allow the task T1 to access the resource R1 in said operation until the task T2 attempts to access the resource R1 and accesses the resource R1 in said operation.

12 (original). The circuit of Claim 11 wherein for at least one resource, each task starts accessing the resource by locking the resource to make it unavailable to any other task, and the task finishes accessing the resource by unlocking the resource.

13 (currently amended). A method for sharing a plurality of resources by a plurality of computer tasks in a continuous operation, the method comprising each of said tasks accessing each of said resources in said operation;

wherein accessing any one of said resources by any one of said tasks comprises:

(1) the task generating a signal indicating that the task is attempting to access the resource;

(2) in response to the operation (1), allowing the task to access the resource if the resource is available to the task, but not allowing the task to access the resource until the resource becomes available to the task;

(3) the task accessing the resource when allowed to access the resource;

wherein for any task T1 of said tasks and any resource R1 of said resources, if the task T1 attempts to access the resource R1 after the task T1 has already finished accessing the resource R1, and at least one other task T2 has not attempted to access the resource R1 after the task T1 has finished accessing the resource R1, then the circuit ~~does not~~ will never allow the task T1 to access the resource R1 in said operation until the task T2 attempts to access the resource R1 and accesses the resource R1 in said operation.

14-29 (canceled).

30 (previously presented). The circuit of Claim 11 wherein each data unit is processed by a single one of the tasks which accesses at least two of said resources to process at least one of the data units.

31 (previously presented). The circuit of Claim 30 wherein said at least two resources are storage areas each of which is to store data unit processing information of multiple data units.

32 (previously presented). The circuit of Claim 31 wherein one of the storage areas is a request FIFO for storing requests to process data units, and another one of the storage areas is a command FIFO for storing commands for processing of the data units;

wherein for each data unit, one of the tasks reads a request from the request FIFO and, if one or more commands are to be written to the command FIFO for the data unit, said one of the tasks writes the one or more commands to the command FIFO.

33 (previously presented). The circuit of Claim 32 wherein each request contains an address of a corresponding data unit.

34 (previously presented). The circuit of Claim 31 wherein at least some of the data units are received over a network, wherein one of the storage areas is a request FIFO for storing requests to process the data units, and another of the storage areas is a status FIFO for storing status information on reception of the data units over the network;

wherein for each data unit, one of the tasks reads a request from the request FIFO and reads the status information from the status FIFO.

35 (previously presented). The method of Claim 13 wherein for at least one resource, each task starts accessing the resource by locking the resource to make it unavailable to any other task, and the task finishes accessing the resource by unlocking the resource.

36 (previously presented). The method of Claim 13 wherein each data unit is processed by a single one of the tasks which accesses at least two of said resources to process at least one of the data units.

37 (previously presented). The method of Claim 36 wherein said at least two resources are storage areas each of which is to store data unit processing information of multiple data units.

38 (previously presented). The method of Claim 37 wherein one of the storage areas is a request FIFO for storing requests to process data units, and another one of the storage areas is a command FIFO for storing commands for processing of the data units;

wherein for each data unit, one of the tasks reads a request from the request FIFO and, if one or more commands are to be written to the command FIFO for the data unit, said one of the tasks writes the one or more commands to the command FIFO.

39 (previously presented). The method of Claim 38 wherein each request contains an address of a corresponding data unit.

40 (previously presented). The method of Claim 37 wherein at least some of the data units are received over a network, wherein one of the storage areas is a request FIFO for storing requests to process the data units, and another of the storage areas is a status FIFO for storing status information on reception of the data units over the network;

wherein for each data unit, one of the tasks reads a request from the request FIFO and reads the status information from the status FIFO.

41 (currently amended). The circuit of Claim [[30]] 31 wherein in each said storage area, the data processing information is associated with an order of data units, the order being the same for each said storage area.

42 (previously presented). The method of Claim 37 wherein in each said storage area, the data processing information is stored in association with an order of data units, the order being the same for each said storage area.

43 (new). The circuit of claim 11 wherein in said operation, at least one of the resources is accessed multiple times by each of said tasks.

44 (new). The circuit of claim 30 wherein in processing each of said data units, the corresponding one of the tasks accesses the resources one after another in a predefined sequence.

45 (new). The method of claim 13 wherein at least one of the resources is accessed multiple times by each of said tasks.

46 (new). The method of claim 37 wherein in processing each of said data units, the corresponding one of the tasks accesses the resources one after another in a predefined sequence.

47 (new). In a multi-tasking computer system comprising a plurality of resources to be shared by a plurality of tasks, a circuit for allocating each resource to the tasks so that after any one of the tasks has finished accessing any one of the resources in

processing a data unit, said one of the tasks does not get access to the same resource until after every other one of the tasks has finished accessing the resource;

wherein accessing any one of said resources by any one of said tasks comprises:

(1) the task attempting to access the resource, wherein attempting to access the resource comprises generating a signal indicating that the task is attempting to access the resource;

(2) in response to the operation (1), the circuit allowing the task to access the resource if the resource is available to the task, the circuit not allowing the task to access the resource until the resource becomes available to the task;

(3) the task accessing the resource when the circuit allows the task to access the resource;

wherein for any task T1 of said tasks and any resource R1 of said resources, if the task T1 attempts to access the resource R1 after the task T1 has already finished accessing the resource R1, and at least one other task T2 has not attempted to access the resource R1 after the task T1 finished accessing the resource R1, then the circuit does not allow the task T1 to access the resource R1 until the circuit obtains an indication of one or more conditions including a condition that every task other than T1 has finished accessing the resource R1 after the task T1 accessed the resource R1, the circuit being responsive to said indication to allow the task T1 to access the resource R1.

48 (new). The circuit of claim 47 wherein in said operation, at least one of the resources is accessed multiple times by each of said tasks.

49 (new). The circuit of claim 47 wherein each data unit is processed by a single one of the tasks which accesses the resources one after another in a predefined sequence in processing the data unit.

50 (new). A method for sharing a plurality of resources by a plurality of computer tasks, the method comprising each of said tasks accessing each of said resources;

wherein accessing any one of said resources by any one of said tasks comprises:

(1) the task generating a signal indicating that the task is attempting to access the resource;

(2) in response to the operation (1), allowing the task to access the resource if the resource is available to the task, but not allowing the task to access the resource until the resource becomes available to the task;

(3) the task accessing the resource when allowed to access the resource;

wherein for any task T1 of said tasks and any resource R1 of said resources, if the task T1 attempts to access the resource R1 after the task T1 has already finished accessing the resource R1, and at least one other task T2 has not attempted to access the resource R1 after the task T1 finished accessing the resource R1, then the circuit does not allow the task T1 to access the resource R1 until the circuit obtains an indication of one or more conditions including a condition that every task other than T1 has finished accessing the resource R1 after the task T1 accessed the resource R1, the circuit being responsive to said indication to allow the task T1 to access the resource R1.

51 (new). The method of claim 50 wherein at least one of the resources is accessed multiple times by each of said tasks.

52 (new). The method of claim 51 wherein each data unit is processed by a single one of the tasks which accesses the resources one after another in a predefined sequence in processing the data unit.